



به نام خدا



فاز اول پروژه کامپایلرها و زبان های برنامه نویسی

پاییز ۹۸

مهلت تحویل: ۴ آبان

در فاز اول پروژه، شما باید به کمک ابزار ANTLR و زبان برنامه نویسی جاوا، برای زبان ACTon (که سند آن را در اختیار دارید) تحلیل گر لغوی و نحوی بنویسید.

۱. تحلیل گر لغوی^۱: برای این بخش از پروژه باید به کمک ابزار ANTLR، تمامی token های مورد نظر خود را مشخص کرده و پیاده سازی کنید.
۲. تحلیل گر نحوی^۲: در این بخش از پروژه، ابتدا با نوشتن قواعد نحوی صحیح، گرامر زبان ACTon را به کمک ANTLR پیاده سازی می کنید. بهتر است برای هر کدام از قواعد گرامر خود نام مناسب انتخاب کنید و سپس با اعمال ورودی های مناسب و تشکیل درخت Parse، گرامر خود را تست کنید. توجه داشته باشید که در این فاز، نیازی نیست که هیچ گونه قاعده معنایی^۳ را پیاده سازی کنید. برای مثال، مواردی از قبیل وجود دو متغیر هم نام، ارجاع به متغیری که تعریف نشده، بررسی RValue و LValue بودن عملوندها، وجود break خارج از حلقه و ... همگی از مواردی هستند که مربوط به تحلیل معنایی هستند و در این فاز نیازی نیست برای آنها اقدامی صورت گیرد. (توجه کنید که مواردی هستند که بیان نشده اند و انتظار می رود شما آنها را رعایت کنید و در تست ها بررسی می شوند) اما گرامر شما باید به نحوی باشد که اولویت عملگرها در آن رعایت شده باشد. همچنین، گرامر شما نباید شامل چپگردی یا ابهام باشد.
۳. در ادامه، پس از نوشتن تحلیلگر لغوی و نحوی، باید به کمک Action هایی که به زبان جاوا می نویسید و به گرامر خود اضافه می کنید، خروجی های زیر را (بر اساس پیمایش preorder در درخت (parse در کنسول چاپ کنید. (هر یک را در یک خط جداگانه چاپ کنید)
- با رسیدن به تعریف هر اکتور، نام اکتور را به صورت زیر (قبل از مشاهده ی دستورات داخل آن) چاپ کنید:

¹ Lexer

² Parser

³ Semantic Rule

ActorDec:ActorName,ParentActorName

در صورتی که یک اکتور parent نداشت، تنها نام آن اکتور را (بدون کاما) چاپ کنید.

- با رسیدن به تعریف هر MsgHandler، به صورت زیر نام آن و نام آرگومان های ورودی آن را چاپ کنید:

MsgHandlerDec:MsgHandlerName,arg1,arg2,...

- با رسیدن به تعریف هر اکتور شناخته شده، نام آن و تایپ آن را به صورت زیر چاپ کنید:

KnownActor:ActorType,ActorName

- با رسیدن به تعریف هر متغیر، نام آن و تایپ آن را به صورت زیر چاپ کنید:

VarDec:VarType,VarName

(شامل تعریف actorvar ها، آرگومان های ورودی هندلرها و متغیرهای محلی)

- در صورت مشاهده یک عملگر، تنها خود عملگر را به صورت زیر چاپ کنید:

Operator:OperatorSymbol

به عنوان مثال برای $x = y + 2$ ، به صورت زیر عملگر را چاپ کنید:

Operator: =

Operator: +

توجه: دقت کنید که لازم نیست عملگرهای [], () و , را چاپ کنید.

- در صورت مشاهده ی دستورات if یا else، به صورت زیر آن ها را چاپ کنید:

Conditional:if یا Conditional:else

- در صورت مشاهده ی دستور for، آن را به صورت زیر نمایش دهید:

Loop:for

- در صورت مشاهده ی فراخوانی MsgHandler، به صورت زیر آن را چاپ کنید:

MsgHandlerCall:ActorInstance,MsgHandlerName

که ActorInstance می تواند self یا sender یا نام Actor ای که هندلر آن فراخوانی شده است باشد. به عنوان مثال برای `a.foo(2)` و `self.bar()`، به ترتیب خروجی های زیر را چاپ کنید:

MsgHandlerCall:a,foo

MsgHandlerCall:self,bar

- در صورت مشاهده‌ی دستور Print، آن را به صورت زیر نمایش دهید:

Built-in:Print

- با رسیدن به تعریف هر نمونه^۴ اکتور در main، آن را به همراه اکتورهای شناخته‌شده اش به صورت زیر چاپ کنید:

ActorInstantiation:ActorType,ActorName,knownactor1,knownactor2,...

به عنوان مثال خروجی تکه کد زیر به صورت زیر است:

```
actor A(5) {
  knownactors {
    A a;
  }

  actorvars {
    int i;
    boolean odd;
  }

  msghandler initial(int i2) {
    i = i2;
    even = true;
  }

  msghandler foo() {
    if(odd == true){
      print(i * i + i);
    }
    odd = !odd;
    i = (i++);
  }

  msghandler bar() {
    a.foo();
  }
}

main {
  A a1(a2):(0);
  A a2(a1):(1);
}
```

⁴ Instance

ActorDec:A
KnownActor:A,a
VarDec:int,i
VarDec:boolean,odd
MsgHandlerDec:initial,i2
Operator:=
Operator:=
MsgHandlerDec:foo
Conditional:if
Operator:==
Built-in:Print
Operator:+
Operator:*
Operator:=
Operator:!
Operator:=
Operator:++
MsgHandlerDec:bar
MsgHandlerCall:a,foo
ActorInstantiation:A,a1,a2
ActorInstantiation:A,a2,a1

توجه: دقت کنید که عبارت‌های قبل از : همگی کلیدواژه هستند و آن‌ها را عیناً چاپ کنید. همچنین، تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خط‌های خالی و فاصله^۵ و ... نیز خودداری کنید.

⁵ White Space

نکات مهم:

- در صورت پیاده سازی گرامر مربوط به ternary expression به صورت صحیح و با رعایت اولویت و نکات ذکر شده در مستند زبان، نمره امتیازی این فاز را دریافت میکنید. دقت شود که این expression تنها برای این فاز است و برای سایر فاز ها، جهت سهولت کارتان حذف میگردد.
- کد خود را به صورت یک فایل StudentID1_StudentID2.g4 آپلود کنید.
- سوالات خود را در فروم درس مطرح نمایید تا دوستانتان نیز از آن ها استفاده کنند.
- دقت کنید که خروجی های شما به صورت خودکار تست می شوند. لذا سعی کنید تا خروجی شما دقیقا همانند خروجی خواسته شده باشد.